IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application. No: 10/749,260
Filed: December 31, 2003
Inventor(s):
Dilip Madhusudan Ranade

Title: CONFLICT RESOLUTION FOR A DISTRIBUTED FILE SHARING SYSTEM

Examiner: Ahn, Sangwoo
Group/Art Unit: 2166

Atty. Dkt. No: 5760-18700

****CERTIFICATE OF E-FILING TRANSMISSION****
I hereby certify that this correspondence is being transmitted via electronic filing to the United States Patent and Trademark Office on the date shown below

B. Noël Kivlin
Printed Name

Signature

September 5, 2007
Date

APPEAL BRIEF

Box: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

Further to the Notice of Appeal filed July 5, 2007, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by VERITAS Operating Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 20330 Stevens Creek Boulevard, Cupertino, CA 95014.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 30-51 are pending in the application. All of the pending claims stand rejected and are the subject of this appeal. A copy of the claims incorporating entered amendments is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

All amendments have been entered. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE INDEPENDENT CLAIMS

Independent claim 30 recites a system comprising a network and a plurality of computing nodes coupled via the network. (*See, for example, network 102 and nodes 110 in FIG. 1; p. 6, line 10*). The plurality of nodes includes a first node operable to create a first file representing a first version of a data object. (*See, for example, node 110 and file replicas 109 of FIG. 2; p. 8, lines 16-17; p. 18, lines 2-4 and lines 7-9*). The first node is further operable to detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object. (*See, for example, block 10 of FIG. 3A; p. 13, line 14*).

In response to detecting the conflict, the first node is operable to modify a tree structure representing the data object to reflect the conflict. Modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object. (*See, for example, block 12 of FIG. 3A; FIG. 3G - branching from Version 2 to Version 3a and Version 3b; p. 13, lines 14-16; p. 16, lines 1-4*). The first replica of the first version of the data object represents the second version of the data object, and the second replica of the first version of the data object represents the third version of the data object. (*See, for example, p. 16, lines 4-7*).

Further in response to detecting the conflict, the first node is operable to create a second file representing the second version of the data object and create a third file representing the third version of the data object. (*See, for example, p. 18, lines 7-11*).

Independent claim 38 recites a method, the method comprising creating a first file representing a first version of a data object. (*See, for example, file replicas 109 of FIG. 2; p. 8, lines 16-17; p. 18, lines 2-4 and lines 7-9*). The method further comprises detecting a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object. (*See, for example, block 10 of FIG. 3A; p. 13, line 14*).

In response to detecting the conflict, the method operates to modify a tree structure representing the data object to reflect the conflict. Modifying the tree structure

comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object. (*See, for example, block 12 of FIG. 3A; FIG. 3G - branching from Version 2 to Version 3a and Version 3b; p. 13, lines 14-16; p. 16, lines 1-4*). The first replica of the first version of the data object represents the second version of the data object, and the second replica of the first version of the data object represents the third version of the data object. (*See, for example, p. 16, lines 4-7*).

Further in response to detecting the conflict, the method operates to create a second file representing the second version of the data object and create a third file representing the third version of the data object. (*See, for example, p. 18, lines 7-11*).

Independent claim 43 recites a computer-readable memory medium comprising program instructions. (*See, for example, FIG. 2 - memory 122 storing executable software; p. 9, lines 20-25; p. 54, lines 7-8*). The program instructions are executable to create a first file representing a first version of a data object. (*See, for example, file replicas 109 of FIG. 2; p. 8, lines 16-17; p. 18, lines 2-4 and lines 7-9*). The program instructions are further executable to detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object. (*See, for example, block 10 of FIG. 3A; p. 13, line 14*).

In response to detecting the conflict, the program instructions are executable to modify a tree structure representing the data object to reflect the conflict. Modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object. (*See, for example, block 12 of FIG. 3A; FIG. 3G - branching from Version 2 to Version 3a and Version 3b; p. 13, lines 14-16; p. 16, lines 1-4*). The first replica of the first version of the data object represents the second version of the data object, and the second replica of the first version of the data object represents the third version of the data object. (*See, for example, p. 16, lines 4-7*).

Further in response to detecting the conflict, the program instructions are executable to create a second file representing the second version of the data object and

create a third file representing the third version of the data object. (*See, for example, p. 18, lines 7-11*).

Independent claim 48 recites a computing node comprising one or more processors (*see, for example, FIG. 2 – node 110 including processor 120; p. 7, lines 12-13*) and a memory storing program instructions (*see, for example, FIG. 2 – node 110 including memory 122 storing executable software; p. 9, lines 20-25*). The program instructions are executable by the one or more processors to create a first file representing a first version of a data object. (*See, for example, file replicas 109 of FIG. 2; p. 8, lines 16-17; p. 18, lines 2-4 and lines 7-9*). The program instructions are further executable by the one or more processors to detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object. (*See, for example, block 10 of FIG. 3A; p. 13, line 14*).

In response to detecting the conflict, the program instructions are executable by the one or more processors to modify a tree structure representing the data object to reflect the conflict. Modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object. (*See, for example, block 12 of FIG. 3A; FIG. 3G - branching from Version 2 to Version 3a and Version 3b; p. 13, lines 14-16; p. 16, lines 1-4*). The first replica of the first version of the data object represents the second version of the data object, and the second replica of the first version of the data object represents the third version of the data object. (*See, for example, p. 16, lines 4-7*).

Further in response to detecting the conflict, the program instructions are executable by the one or more processors to create a second file representing the second version of the data object and create a third file representing the third version of the data object. (*See, for example, p. 18, lines 7-11*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Section 101 Rejections

Claims 30-37 and 43-47 stand rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

Section 102 Rejections

Claims 30-32, 34-40, 42-45, and 47-50 stand rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 4,853,843 to Ecklund (hereinafter "Ecklund").

Section 103 Rejections

Claims 33, 41, 46, and 51 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Ecklund in view of U.S. Patent No. 6,003,034 to Tuli (hereinafter "Tuli").

VII. ARGUMENT

Section 101 Rejections

Claims 30-37 and 43-47 stand rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter. Appellant respectfully traverses these rejections.

Regarding claims 30-37, the Examiner states that the claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 U.S.C. 101. Appellant disagrees. Claim 30 recites a system comprising a plurality of computing nodes, wherein the plurality of computing nodes includes a first node operable to perform various features as set forth in the remainder of the claim. Appellant respectfully submits that the plurality of computing nodes including the first node constitutes a machine or a manufacture within the meaning of 35 U.S.C. 101 and that claims 30-37 are directed to statutory subject matter.

Regarding claims 43-47, the Examiner asserts:

> "On page 54 lines 9 - 10 of the instant specification, applicant has provided evidence that applicant intends the 'medium' to include signals. As such, the claim is drawn to a form of energy."

Appellant respectfully submits that although the cited portion of the specification refers to a carrier medium, which is described as including memory media as well as transmission media or signals, claims 43-47 specifically recite a memory medium. Accordingly, these claims are believed to be statutory.

Independent claims 30, 38, 43, and 48

Claims 30-32, 34-40, 42-45, and 47-50 stand rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 4,853,843 to Ecklund (hereinafter "Ecklund"). Appellant respectfully traverses these rejections.

Ecklund relates generally to a system for merging virtual partitions of a distributed database. A database system provides a plurality of separate virtual partitions,

each storing separate instances of an initial database. The database comprises an initial set of versions of data objects, each version of a particular data object, other than a first version of the data object, being created by modifying an existing version of the data object. (Col. 2, lines 57 – 63). Each virtual partition independently executes group updates. A group update may, for example, add a new version of a data object to a path. Each virtual partition maintains a separate change list describing all group updates that it executes. (Col. 3, lines 16-24). Each virtual partition may provide a merged database reflecting changes to the initial database resulting from all group updates described by change lists maintained by the separate virtual partitions. (Col. 3, lines 25-29). For a respective virtual partition, forming the merged database includes adding additional data object versions to alternate paths of the data objects. (Col. 3, lines 45-50).

Appellant respectfully submits that Ecklund does not teach all of the limitations recited in the independent claims. For example, claim 30 recites, in pertinent part, a first node operable to create a first file representing a first version of a data object. As discussed above, Ecklund teaches data objects stored in a distributed database. As well known to those skilled in the art, a data object stored in a database is not at all the same as a file. Ecklund nowhere teaches creating a first file representing a first version of a data object, as recited in claim 30.

Ecklund also does not teach the recited limitations of creating "a second file representing the second version of the data object" and "a third file representing the third version of the data object." As described above, Ecklund teaches adding additional data object versions to alternate paths of data objects in a database. With respect to the above claim limitations, the Examiner asserts that, "each version is a data object or 'copies' of the original version, which essentially means it could be a file stored in a storage." However, there is no basis whatsoever in Ecklund for the Examiner's assertion that each version "could be a file stored in a storage". Instead, Ecklund clearly teaches that each version is a version of a data object stored in a database. As noted above, a data object stored in a database is not at all the same as a file.

Thus, Appellant respectfully submits that claim 30, and the claims respectively dependent thereon, are patentably distinct over Ecklund for at least the reasons set forth above. Inasmuch as the independent claims 38, 43, and 48 recite similar limitations as

those discussed above, Appellant respectfully submits that these claims, and the claims respectively dependent thereon, are also patentably distinct over Ecklund.

Claims 31, 39, 44, and 49

Claim 31 adds to claim 30 the additional limitations of, "wherein creating the second file and the third file comprises creating the second file and the third file in a common directory." As discussed above, Ecklund does not teach creating the second file and the third file, in combination with the other limitations recited in claim 30. Therefore, Ecklund does not, and cannot, teach that creating the second file and the third file comprises creating the second file and the third file in a common directory.

In the rejection of claim 31, the Examiner cites FIG. 12 and Col. 2, lines 63-68. FIG. 12 simply illustrates forming a merged database from two group updates formed in different partitions (Col. 50, lines 5-7). As per Col. 2, lines 63-68, Ecklund teaches here that:

> A data base system provides a plurality of separate virtual partitions, each storing separate instances of an initial data base. The data base comprises an initial set of versions of data objects, each version of a particular data object, other than a first version of the data object, being created by modifying an existing version of the data object. The initial data base also includes an initial set of directory data associated with each data object, the directory data identifying non-overlaping "paths of descendancy" for the associated data object, wherein each path of descendancy comprises sequentially created versions of the data object. Each data object version included in a path, except a first data object version of the path, directly descends from a last created data object version of the path. The directory data classifies each path as one of "principal" and "alternate", and classifies each version of each path as being one of "current" and "non-current". However, one and only one path of each object is classified as "principal" and one and only one version of each path is classified as "current".

This teaching has nothing whatsoever to do with creating two files in a common directory. Thus, Appellant respectfully submits that claim 31 is separately patentable over Ecklund since Ecklund does not teach the further limitations recited therein.

Inasmuch as claims 39, 44, and 49 recite similar limitations as claim 31, Appellant respectfully submits that these claims are also separately patentable over Ecklund.

Claims 32, 40, 45, and 50

Claim 32 adds to claim 30 the additional limitations of, "wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file." As discussed above with respect to claim 31, Ecklund does not teach creating the second file and the third file in a common directory, in combination with the other recited claim limitations. Thus, Ecklund does not, and cannot, teach creating the second file and the third file in a common directory with the first file. Appellant thus respectfully submits that claim 32 is separately patentable over Ecklund since Ecklund does not teach the further limitations recited therein.

Inasmuch as claims 40, 45, and 50 recite similar limitations as claim 31, Appellant respectfully submits that these claims are also separately patentable over Ecklund.

Claim 36

Claim 35 adds to claim 30 the additional limitations of, "wherein the conflict between the two replicas is caused by update operations that update the two replicas." Claim 36 adds to claim 35 the additional limitations of, "wherein the update operations that update the two replicas comprise concurrent update operations." Ecklund teaches that corresponding versions of a data object in different virtual partitions are updated independently of each other. (Col. 3, lines 16-19). However, Ecklund nowhere teaches that the versions are updated by concurrent update operations. Updating versions of a data object in different virtual partitions independently of each other is not the same as updating two replicas by concurrent update operations. Appellant thus respectfully submits that claim 36 is separately patentable over Ecklund since Ecklund does not teach the further limitations recited therein.

Claims 33, 41, 46, and 51

Claims 33, 41, 46, and 51 were rejected under 35 U.S.C. 103(a) as being unpatentable over Ecklund in view of Tuli. Appellant respectfully traverses these rejections.

Claim 33 adds to claim 30 the additional limitations of,

wherein the first file has a first name;
wherein creating the second file comprises creating the second file with a second name based on the first name; and
wherein creating the third file comprises creating the third file with a third name based on the first name.

Appellant respectfully submits that Ecklund and Tuli, taken either singly or in combination, do not teach these limitations in combination with the limitations of claim 30. The Examiner states that, "Tuli teaches creating a name of a file based on the name of another file". However, as discussed above, Ecklund does not teach creating the first file, the second file, or the third file recited in claim 33. Thus, Ecklund requires more of Tuli than simply creating a name of a file based on the name of another file.

Appellant also respectfully reminds the Board that, "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)," as stated in the MPEP §2142.

As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a

combination is no more than hindsight without an initial suggestion to make the combination.

Applicant respectfully submits that there is no clear and particular teaching or suggestion in the prior art for combining Ecklund and Tuli. The Examiner states that:

> "At the time of the present invention, it would have been obvious to a person of ordinary skill in the data processing art to combine the two references because the combination would have enabled a simplified method of file management which assigns many related attributes to any file created, resulting in an extremely efficient and resourceful database system with numerous avenues to locate any file."

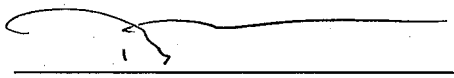
However, the motivation given by the Examiner amounts to a broad conclusory statement and does not amount to a clear and particular teaching or suggestion to combine the references, as would be required to form a case of *prima facie* obviousness. Appellant also disagrees that the combination of Ecklund and Tuli would result in the advantages asserted by the Examiner and respectfully submits that there is no teaching or suggestion in the prior art that would motivate one skilled in the art to combine the references.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of claims 30-51 were erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-18700.

Respectfully submitted,



B. Noël Kivlin
Reg. No. 33,929
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: September 5, 2007

IX. CLAIMS APPENDIX

The following lists the claims as incorporating entered amendments, and as on appeal.

1 – 29. (Canceled)

30. (Previously Presented) A system comprising:

a network; and

a plurality of computing nodes coupled via the network;

wherein the plurality of nodes includes a first node operable to:

create a first file representing a first version of a data object;

detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;

in response to detecting the conflict:

modify a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;

create a second file representing the second version of the data object; and

create a third file representing the third version of the data object.

31. (Previously Presented) The system of claim 30,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory.

32. (Previously Presented) The system of claim 31,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file.

33. (Previously Presented) The system of claim 30,

wherein the first file has a first name;

wherein creating the second file comprises creating the second file with a second name based on the first name; and

wherein creating the third file comprises creating the third file with a third name based on the first name.

34. (Previously Presented) The system of claim 30,

wherein the first replica of the first version of the data object is stored on a second node and the second replica of the first version of the data object is stored on a third node.

35. (Previously Presented) The system of claim 30,

wherein the conflict between the two replicas is caused by update operations that update the two replicas.

36. (Previously Presented) The system of claim 35,

wherein the update operations that update the two replicas comprise concurrent update operations.

37. (Previously Presented) The system of claim 30,

wherein the conflict between the two replicas is caused by the two replicas being independently updated in different network partitions.

38. (Previously Presented) A method comprising:

creating a first file representing a first version of a data object;

detecting a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;

in response to detecting the conflict:

modifying a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;

creating a second file representing the second version of the data object; and

creating a third file representing the third version of the data object.

39. (Previously Presented) The method of claim 38,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory.

40. (Previously Presented) The method of claim 39,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file.

41. (Previously Presented) The method of claim 38,

wherein the first file has a first name;

wherein creating the second file comprises creating the second file with a second name based on the first name; and

wherein creating the third file comprises creating the third file with a third name based on the first name.

42. (Previously Presented) The method of claim 38,

wherein the first replica of the first version of the data object is stored on a first node and the second replica of the first version of the data object is stored on a second node.

43. (Previously Presented) A computer-readable memory medium comprising program instructions executable to:

create a first file representing a first version of a data object;

detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;

in response to detecting the conflict:

modify a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;

create a second file representing the second version of the data object; and

create a third file representing the third version of the data object.

44. (Previously Presented) The computer-readable memory medium of claim 43,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory.

45. (Previously Presented) The computer-readable memory medium of claim 44,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file.

46. (Previously Presented) The computer-readable memory medium of claim 43,

wherein the first file has a first name;

wherein creating the second file comprises creating the second file with a second name based on the first name; and

wherein creating the third file comprises creating the third file with a third name based on the first name.

47. (Previously Presented) The computer-readable memory medium of claim 43,

wherein the first replica of the first version of the data object is stored on a first node and the second replica of the first version of the data object is stored on a second node.

48. (Previously Presented) A computing node comprising:

one or more processors; and

a memory storing program instructions;

wherein the one or more processors are operable to execute the program instructions to:

create a first file representing a first version of a data object;

detect a conflict between a first replica of the first version of the data object and a second replica of the first version of the data object;

in response to detecting the conflict:

modify a tree structure representing the data object to reflect the conflict, wherein modifying the tree structure comprises adding information to the tree structure representing a branching from the first version of the data object to a second version of the data object and a third version of the data object, wherein the first replica of the first version of the data object represents the second version of the data object and the second replica of the first version of the data object represents the third version of the data object;

create a second file representing the second version of the data object; and

create a third file representing the third version of the data object.

49. (Previously Presented) The computing node of claim 48,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory.

50. (Previously Presented) The computing node of claim 49,

wherein creating the second file and the third file comprises creating the second file and the third file in a common directory with the first file.

51. (Previously Presented) The computing node of claim 48,

wherein the first file has a first name;

wherein creating the second file comprises creating the second file with a second name based on the first name; and

wherein creating the third file comprises creating the third file with a third name based on the first name.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.